IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No. 10/686,820 Group Art Unit 2617

Filing Date: October 16, 2003 Examiner: Wesley L. Kim

Inventors: Gabriel, et al.

Title: System and method for sending SMS and text messages

DECLARATION OF MANNY MANIMTIM GABRIEL

I, Manny Manimtim Gabriel, having first been warned that willful false statements and the like are punishable by fine or imprisonment, or both, and that a false declaration may jeopardize the validity of this application or any patent issuing thereon, make the following Declaration of my own knowledge:

1. I am a resident of Guam, and am a named inventor on U.S. Patent Application No. 10/686,820. Prior to December, 2000, I had given some thought to the concept of short messaging service ("SMS") as then implemented for use in cellular telephones. My interest was based upon the fact that, being located in Guam, I frequently found it difficult to communicate with family and friends elsewhere in the world, and the SMS feature of cellular telephones appeared to be a facility that held promise for being able to provide a nearly instantaneous, inexpensive communications medium between persons in geographically remote locations. I conceived the invention together with my brother, co-inventor Leandro Manimtim Gabriel, in January, 2001. Leandro, who also lives in Guam, and I had been visiting our parents in the Philippines in December, 2000 and January, 2001. While there, we saw the widespread use of short messaging service ("SMS") in the Philippines.

2. Upon returning from vacation in the Philippines in January 2001 we wanted to find a way to continue to communicate with our friends and family in the Philippines using some form of instant messaging. We wished to be able to communicate from Guam with our parents in the Philippines using the now very popular in the Philippines SMS. In January, 2001, however, there were no compatible SMS Network Providers in Guam. At that time, in January, 2001, we conceived the idea of integrating internet communications with mobile communications to route SMS messages between different cellular systems, and between geographically remote areas such as Guam and the Philippines.

3. We both had full time jobs, but were able to spend around 3 to 4 hours every day after work and on weekends doing research and developing the system. After some research, programming, configuration and testing with different hardware, in late April of 2001, the first messages were sent via a server in our home in Guam connected via IP to a server in the Philippines that was physically connected to a mobile phone via a data cable. Our father was the recipient of our first message. When he replied with a message of his own, it went back through the mobile phone connected to the server in the Philippines, and was then forwarded back to our server in our home in Guam where we received it.

4. After some tweaking, our first working prototype of SMS via the Web was used consistently to communicate with our parents in the Philippines around June, 2001. On June 21, 2001, we purchased IPWorks, a Visual Basic Control, to use with our software to improve stability and performance for an improved version of SMS that we began using in July 2001.

5. In late July we began to concentrate on the Client Software and the Web Interface to allow for the system to accommodate multiple users. By August 2001 we had some beta testers in Guam using our system to communicate with their friends and families in the Philippines.

6. In September, 2001, Guam's first compatible GSM network provider (Hafatel) came open for business, and by early 2002 we were able to use that network for our continuing development of the system incorporating cell phones. We began working on the option of utilizing their mobile phones with our system to create an SMS that operated phone-to-phone via the Web. Messages utilizing this option would travel from the user's mobile phone to a mobile phone connected via data cable to a server in Guam, would then be converted to IP and travel over the internet to a server in the Philippines, and would then be reconverted and sent via data cable to a mobile phone in the Philippines. This improvement gave us the capability to utilize our system without having to physically be at a computer with internet access.

7. As we improved the invention to include phone-to-phone SMS using the internet, we also found it necessary to develop software routing and addressing algorithms suitable to permit phone users to conveniently address e-mails to other phone without having to enter large amounts of alphanumeric data. Throughout 2002, we evolved various software programs and modules to provide the necessary address and communications interfaces to permit SMS messages to flow between cellular systems, and individual phones, via the internet.

8. On October 17, 2002, we filed our provisional patent application under App. No. 60/419804. Thereafter, we continued to work on general enhancements and improvements for a multiple-user SMS system.

9. I have attached a number of exhibits to this Declaration to show the dates upon which certain files that we used in the development of the invention had been last modified. Our software code was written and modified daily, and most of our files continued to be modified after we started using them. There are a few files, however, that were not modified later on, and that provide a definite "last date" of modification. These are noted below.

10. Attached hereto as Exhibit A is a file entitled "GTModule.bas" - a file that we created in Visual Basic and last updated on August 29, 2001. This file represents an array definition file. It defines the structure of an array that would hold information necessary to identify currently connected SMS devices. This array definition is used to provide addresses and other connection information needed for sending SMS messages to connected SMS devices using our invention.

11. Attached hereto as Exhibit B is a file entitled "Module1.bas" – a file containing some publicly available code that I last modified to use with our product on August 29, 2001.

12. Attached hereto as Exhibits C-1 through C-7 are screen images taken from an old computer that we were then using to develop the invention. Although our development activities were thereafter moved to another development platform, the computer shown in Exhibits C-1

through C-7 was being used during the summer months of 2001, and the files shown in the image were then existing upon that computer.

13. Exhibit C-1 is a screen image of computer files in a directory entitled "SMS." With the exception of a file (RagnoSMS1Beta.vbw) last modified on April 9, 2000, the files in the SMS directory show a last modified date of March 1, 2001. Most of these files were created and developed in conjunction with our development of our invention following our conception of the invention. Two subdirectories are also found in SMS, one of which is "SMS Services." Although SMS Services was created on August 26, 2002, it contains files that were created at an earlier date, and that were copied or moved to SMS Services at some later time.

14. Exhibit C-2 shows the files in a directory entitled "SMS Service." These files were all created and developed in connection with developing the invention, and were last modified on May 15, 2001.

15. Exhibit C-3 shows the files in a directory entitled "SMSDevice." One of these files, "Module1.bas" (Exhibit B to this Declaration) was used in the development of the invention, and was last modified on August 16, 2001. Another file, "default.script" was last modified on September 3, 2001. All files in the SMSDevice directory were used in the development of the invention.

16. Exhibit C-4 shows the files in a directory called "SMSServer." Two files in this directory, "GTModule.bas" and "Module1.bas" were last modified on August 29, 2001. This is

the same file as is also found in the SMSDevice directory, but was accessed, and possibly modified, more recently than the file found in the SMSDevice directory. There is also a file, "ObjectIDbarcode.gdb" that was last modified on September 2, 2001. All files in this directory were used in the development of the invention.

17. Exhibit C-5 shows the contents of a directory entitled "SMSProc." All files in this directory were used in the development of the invention, and all have a last modification date of on or before August 4, 2001.

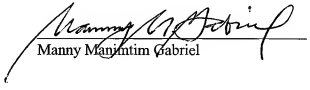
18. Exhibit C-6 shows the contents of a directory called "SMS Edit." With one exception, these files were last modified on January 22, 2001, and were created in connection with our development of our invention.

19. Exhibit C-7 is an image of an internet site from which we purchased and downloaded development software. The last item on the list shows that we downloaded an application entitled IPWorks VB Edition V5.0 on June 29, 2001. This application was used in some of our development work on our invention.

20. These exhibits, and the dates of last modification of the files shown on them, confirm and support my recollection that the invention claimed in our patent application was conceived at least as early as January 1, 2001, and was being developed throughout the time from that date until October 16, 2002, when we filed our provisional patent application. After that date, we continued to work on general enhancements and improvements to the system, and those

enhancements have been directed toward making the system more user-friendly while automating the subscription process and means for adding new addresses to a user's list of persons who could be contacted using the system.

Dated: November 06, 2006



Manny Manimtim Gabriel

EXHIBIT A

```
Attribute VB_Name = "GTModule"
Type GTUsers
    UserID As String
    IsAuthenticated As Boolean
End Type

Type GTDevices
    DeviceID As String
    DeviceType As String
    DeviceVersion As String
    DeviceArea As String
    DeviceLocation As String
    IsAuthenticated As Boolean
End Type

Global User(0 To 9) As GTUsers
Global Device(0 To 49) As GTDevices
```

EXHIBIT B

```
Attribute VB_Name = "INIModule"
'Not thouroughly commented, comments desribe what each function does.
'Please see Form1 code to see how to call each function

Option Explicit
'APIs to access INI files and retrieve data
Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long
Declare Function WritePrivateProfileString& Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal AppName$, ByVal KeyName$, ByVal keydefault$, ByVal FileName$)

Function GetKeyVal(ByVal FileName As String, ByVal Section As String, ByVal Key As String)
'Returns info from an INI file
Dim RetVal As String, Worked As Integer
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
RetVal = String$(255, 0)
Worked = GetPrivateProfileString(Section, Key, "", RetVal, Len(RetVal), FileName)
If Worked = 0 Then
    GetKeyVal = ""
Else
    GetKeyVal = Left(RetVal, InStr(RetVal, Chr(0)) - 1)
End If
End Function

Function AddToINI(ByVal FileName As String, ByVal Section As String, ByVal Key As String, ByVal KeyValue As String) As Integer
'Add info to an INI file
'Function returns 1 if successful and 0 if unsuccessful
WritePrivateProfileString Section, Key, KeyValue, FileName
AddToINI = 1
End Function

Function DeleteSection(ByVal FileName As String, ByVal Section As String) As Integer
'Delete an entire section and all it's keys from a given INI file
'Function returns 1 if successful and 0 if unsuccessful
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. ~(DeleteSection)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
WritePrivateProfileString Section, vbNullString, vbNullString, FileName
DeleteSection = 1
End Function

Function DeleteKey(ByVal FileName As String, ByVal Section As String, ByVal Key As String) As Integer
'Delete a key from an INI file
'Function returns 1 if successful and 0 if unsuccessful
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. ~(DeleteKey)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
If Not KeyExists(FileName, Section, Key) Then MsgBox "Key, " & Key & ", Not Found. ~(DeleteKey)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Key Not Found.": Exit Function
WritePrivateProfileString Section, Key, vbNullString, FileName
DeleteKey = 1
End Function

Function DeleteKeyValue(ByVal FileName As String, ByVal Section As String, ByVal Key As String) As Integer
'Delete a key's value from an INI file
'Function returns 1 if successful and 0 if unsuccessful
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. ~(DeleteKeyValue)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
If Not KeyExists(FileName, Section, Key) Then MsgBox "Key, " & Key & ", Not Found. ~(DeleteKeyValue)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Key Not Found.": Exit Function
WritePrivateProfileString Section, Key, "", FileName
DeleteKeyValue = 1
End Function

Public Function TotalSections(ByVal FileName As String) As Long
'Returns the total number of sections in a given INI file
Dim Counter As Integer
Dim InputData As String
```

```
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
Open FileName For Input As #1

Do While Not EOF(1)
   Line Input #1, InputData
   If IsSection(InputData) Then Counter = Counter + 1
Loop
Close #1
TotalSections = Counter
End Function

Public Function TotalKeys(ByVal FileName As String) As Long
'Returns the total number of keys in a given INI file
Dim Counter As Integer
Dim InputData As String
Dim Looper As Integer
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
Open FileName For Input As #2

Do While Not EOF(2)
   Line Input #2, InputData
   If IsKey(InputData) Then Counter = Counter + 1
Loop
Close #2
TotalKeys = Counter
End Function

Public Function NumKeys(ByVal FileName As String, ByVal Section As String) As Integer
'Returns the total number of keys in 1 given section.
Dim Counter As Integer
Dim InputData As String
Dim Looper As Integer
Dim InZone As Boolean
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. -(NumKeys)" & vbCrLf & "Verify spelling and
capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
InZone = False
Open FileName For Input As #3

Do While Not EOF(3)
   Line Input #3, InputData
   If InZone Then
      If IsSection(InputData) Or EOF(3) Then
         If EOF(3) Then
            NumKeys = Counter + 1
            Exit Do
         Else
            NumKeys = Counter
            Exit Do
         End If
      Else
         If IsKey(InputData) Then Counter = Counter + 1
      End If
   Else
      If InputData = "[" & Section & "]" Then
         InZone = True
      End If
   End If
Loop
Close #3
End Function

Public Function RenameSection(ByVal FileName As String, ByVal SectionName As String, ByVal NewSectionName As String) As Integer
'Renames a section in a given INI file.
'Function returns 1 if successful and 0 if unsuccessful
Dim TopKeys As String
Dim BotKeys As String
Dim Looper As Integer
Dim InputData As String
Dim InZone As Boolean
```

```
Dim Key1 As String, Key2 As String
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, SectionName) Then MsgBox "Section, " & SectionName & ", Not Found. ~(RenameSection)" & vbCrLf &
"Verify spelling and capitilization is correct. Case-sensative.", vbInformation, "Section Not Found.": RenameSection = 0: Exit Function
If SectionExists(FileName, NewSectionName) Then MsgBox NewSectionName & " allready exists. ~(RenameSection)", vbInformation,
"Duplicate Section": RenameSection = 0: Exit Function
Open FileName For Input As #4

Do While Not EOF(4)
  Line Input #4, InputData
  If InZone Then
    If BotKeys = "" Then BotKeys = InputData Else BotKeys = BotKeys & vbCrLf & InputData
    If EOF(4) Then
      Close #4
      Kill FileName
      Open FileName For Append As #5
      If TopKeys <> "" Then Print #5, TopKeys
      Print #5, "[" & NewSectionName & "]" & vbCrLf & BotKeys
      Close #5
      RenameSection = 1
      Exit Function
    End If
  Else
    If InputData = "[" & SectionName & "]" Then
      InZone = True
    Else
      If TopKeys = "" Then TopKeys = InputData Else TopKeys = TopKeys & vbCrLf & InputData
    End If
  End If
Loop
Close #4
End Function

Public Function RenameKey(ByVal FileName As String, ByVal Section As String, ByVal KeyName As String, ByVal NewKeyName As String)
As Integer
'Renames a key in a given INI file
'Function returns 1 if successful and 0 if unsuccessful
Dim KeyVal As String
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": RenameKey = 0: Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. ~(RenameKey)" & vbCrLf & "Verify spelling and
capitilization is correct. Case-sensative.", vbInformation, "Section Not Found.": RenameKey = 0: Exit Function
If Not KeyExists(FileName, Section, KeyName) Then MsgBox "Key, " & KeyName & ", Not Found. ~(RenameKey)" & vbCrLf & "Verify
spelling and capitilization is correct. Case-sensative.", vbInformation, "Key Not Found.": RenameKey = 0: Exit Function
If KeyExists(FileName, Section, NewKeyName) Then MsgBox NewKeyName & " allready exists in the section, " & Section & ".
~(RenameKey)", vbInformation, "Duplicate Key.": RenameKey = 0: Exit Function
KeyVal = GetKeyVal(FileName, Section, KeyName)
DeleteKey FileName, Section, KeyName
AddToINI FileName, Section, NewKeyName, KeyVal
RenameKey = 1
End Function

Public Function GetKey(ByVal FileName As String, ByVal Section As String, ByVal KeyIndexNum As Integer) As String
'This function returns the name of a key which is identified by it's IndexNumber.
'The Section is identified as Text - GetKey2 identifies Section by it's IndexNumber
'IndexNumbers begin at 0 and increment up
Dim Counter As Integer
Dim InputData As String
Dim InZone As Boolean
Dim Looper As Integer
Dim KeyName As String
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. ~(GetKey)" & vbCrLf & "Verify spelling and
capitilization is correct. Case-sensative.", vbInformation, "Section Not Found.": Exit Function
If NumKeys(FileName, Section) - 1 < KeyIndexNum Then MsgBox KeyIndexNum & ", not a valid Key Index Number. ~(GetKey)",
vbInformation, "Invalid Index Number.": Exit Function

Counter = -1
Open FileName For Input As #6
Do While Not EOF(6)
  Line Input #6, InputData
```

```
    If InZone Then
        If IsKey(InputData) Then
            Counter = Counter + 1
            If Counter = KeyIndexNum Then
                For Looper = 1 To Len(InputData)
                    If Mid(InputData, Looper, 1) = "=" Then
                        GetKey = KeyName
                        Exit Do
                    Else
                        KeyName = KeyName & Mid(InputData, Looper, 1)
                    End If
                Next Looper
            End If
        End If
    Else
        If InputData = "[" & Section & "]" Then InZone = True
    End If
Loop
Close #6
End Function

Public Function GetKey2(ByVal FileName As String, ByVal SectionIndexNum As Integer, ByVal KeyIndexNum As Integer) As String
'This function returns the name of a key which is identified by it's IndexNumber.
'The Section is identified by it's IndexNumber
'IndexNumbers begin at 0 and increment up
Dim Counter As Integer
Dim Counter2 As Integer
Dim InputData As String
Dim InZone As Boolean
Dim Looper As Integer
Dim KeyName As String
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If TotalSections(FileName) - 1 < SectionIndexNum Then MsgBox SectionIndexNum & ", not a valid Section Index Number. ~(GetKey2)", vbInformation, "Invalid Index Number.": Exit Function
If NumKeys(FileName, GetSection(FileName, SectionIndexNum)) - 1 < KeyIndexNum Then MsgBox KeyIndexNum & ", not a valid Key Index Number. ~(GetKey2)", vbInformation, "Invalid Index Number.": Exit Function
Counter = -1
Counter2 = -1
Open FileName For Input As #7
Do While Not EOF(7)
    Line Input #7, InputData
    If InZone Then
        If IsKey(InputData) Then
            Counter = Counter + 1
            If Counter = KeyIndexNum Then
                For Looper = 1 To Len(InputData)
                    If Mid(InputData, Looper, 1) = "=" Then
                        GetKey2 = KeyName
                        Exit Do
                    Else
                        KeyName = KeyName & Mid(InputData, Looper, 1)
                    End If
                Next Looper
            End If
        End If
    Else
        If IsSection(InputData) Then Counter2 = Counter2 + 1
        If Counter2 = SectionIndexNum Then InZone = True
    End If
Loop
Close #7
End Function

Public Function GetSection(ByVal FileName As String, ByVal SectionIndexNum As Integer) As String
'Returns a section name which is identified by it's indexnumber
'IndexNumbers begin at 0 and increment up
Dim InputData As String
Dim Counter As Integer
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
```

```
If TotalSections(FileName) - 1 < SectionIndexNum Then MsgBox SectionIndexNum & ", not a valid Section Index Number. -(GetSection)",
vbInformation, "Invalid Index Number.": Exit Function
Counter = -1
Open FileName For Input As #8
Do While Not EOF(8)
   Line Input #8, InputData
   If IsSection(InputData) Then
      Counter = Counter + 1
      InputData = Right(InputData, Len(InputData) - 1)
      InputData = Left(InputData, Len(InputData) - 1)
      If Counter = SectionIndexNum Then GetSection = InputData: Exit Do
   End If
Loop
Close #8
End Function

Public Function IsKey(ByVal TextLine As String) As Boolean
'This function determines whether or not a line of text is a valid Key (ex. "This=key")
'This returns True or False
Dim Looper As Integer
For Looper = 1 To Len(TextLine)
   If Mid(TextLine, Looper, 1) = "=" Then IsKey = True: Exit Function
Next Looper
IsKey = False
End Function

Public Function IsSection(ByVal TextLine As String) As Boolean
'This function determines whether or not a line of text is a
'valid section (ex. "[section]")
'This return's True or False
Dim FirstChar As String, LastChar As String
If TextLine = "" Then Exit Function
FirstChar = Mid(TextLine, 1, 1)
LastChar = Mid(TextLine, Len(TextLine), 1)
If FirstChar = "[" And LastChar = "]" Then IsSection = True Else IsSection = False
End Function

Public Function KeyExists(ByVal FileName As String, ByVal Section As String, ByVal Key As String) As Boolean
'This function determines if a key exists in a given section
'The Section is identified as Text - KeyExists2 identifies Section by its IndexNumber
'This returns True or False
Dim InZone As Boolean
Dim InputData As String
Dim Looper As Integer
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. -(KeyExists)" & vbCrLf & "Verify spelling and
capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
Open FileName For Input As #9
Do While Not EOF(9)
   Line Input #9, InputData
   If InZone Then
      If IsKey(InputData) Then
         If Left(InputData, Len(Key)) = Key Then
            KeyExists = True
            Exit Do
         End If
      ElseIf IsSection(InputData) Then
         KeyExists = False
         Exit Do
      End If
   Else
      If InputData = "[" & Section & "]" Then InZone = True
   End If
Loop
Close #9
End Function

Public Function KeyExists2(ByVal FileName As String, ByVal SectionIndexNum As Integer, ByVal Key As String) As Boolean
'This function determines if a key exists in a given section
'The Section is identified by its IndexNumber
```

```
'IndexNumbers begin at 0 and increment up
'This returns True or False
Dim InZone As Boolean
Dim InputData As String
Dim Looper As Integer
Dim Counter As Integer
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If TotalSections(FileName) - 1 < SectionIndexNum Then MsgBox SectionIndexNum & ", not a valid Section Index Number. -(KeyExists2)", vbInformation, "Invalid Index Number.": Exit Function
Counter = -1
Open FileName For Input As #10
Do While Not EOF(10)
   Line Input #10, InputData
   If InZone Then
      If IsKey(InputData) Then
         If Left(InputData, Len(Key)) = Key Then
            KeyExists2 = True
            Exit Do
         End If
      ElseIf IsSection(InputData) Then
         KeyExists2 = False
         Exit Do
      End If
   Else
      If IsSection(InputData) Then Counter = Counter + 1
      If Counter = SectionIndexNum Then InZone = True
   End If
Loop
Close #10
End Function

Public Function SectionExists(ByVal FileName As String, ByVal Section As String)
'This determines if a section exists in a given INI file
'This returns True or False
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
Dim InputData As String
Open FileName For Input As #11
Do While Not EOF(11)
   Line Input #11, InputData
   If "[" & Section & "]" = InputData Then SectionExists = True: Exit Do
   SectionExists = False
Loop
Close #11
End Function

Public Function GetSectionIndex(ByVal FileName As String, ByVal Section As String) As Integer
'This function is used to get the IndexNumber for a given Section
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. -(GetSectionIndex)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
Dim InputData As String
Dim Counter As Integer
Counter = -1
Open FileName For Input As #12
Do While Not EOF(12)
   Line Input #12, InputData
   If IsSection(InputData) Then Counter = Counter + 1
   If "[" & Section & "]" = InputData Then GetSectionIndex = Counter
Loop
Close #12
End Function

Public Function GetKeyIndex(ByVal FileName As String, ByVal Section As String, ByVal Key As String) As Integer
'This function returns the IndexNumber of a key in a given Section
'The Section is identified as Text - GetKeyIndex2, Section is
'identified by it's IndexNumber
'IndexNumbers start at 0 and increment up
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If Not SectionExists(FileName, Section) Then MsgBox "Section, " & Section & ", Not Found. -(GetKeyIndex)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Section Not Found.": Exit Function
```

```
If Not KeyExists(FileName, Section, Key) Then MsgBox "Key, " & Key & ", Not Found. -(GetKetIndex)" & vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Key Not Found.": Exit Function
Dim InputData As String
Dim InZone As Boolean
Dim Counter As Integer
Counter = -1
Open FileName For Input As #13
Do While Not EOF(13)
   Line Input #13, InputData
   If InZone Then
      If IsKey(InputData) Then
         Counter = Counter + 1
         If Left(InputData, Len(Key)) = Key Then
            GetKeyIndex = Counter
            Exit Do
         End If
      ElseIf IsSection(InputData) Then
         Exit Do
      End If
   Else
      If "[" & Section & "]" = InputData Then InZone = True
   End If
Loop
Close #13
End Function

Public Function GetKeyIndex2(ByVal FileName As String, ByVal SectionIndexNum As Integer, ByVal Key As String) As Integer
'This function returns the IndexNumber of a key in a given Section
'The Section is identified by it's IndexNumber
'IndexNumbers start at 0 and increment up
If Dir(FileName) = "" Then MsgBox FileName & " not found.", vbCritical, "File Not Found": Exit Function
If TotalSections(FileName) - 1 < SectionIndexNum Then MsgBox SectionIndexNum & ", not a valid Section Index Number. -(GetKeyIndex2)",
vbInformation, "Invalid Index Number.": Exit Function
If Not KeyExists(FileName, GetSection(FileName, SectionIndexNum), Key) Then MsgBox "Key, " & Key & ", Not Found. -(GetKetIndex2)" &
vbCrLf & "Verify spelling and capitilization is correct.  Case-sensative.", vbInformation, "Key Not Found.": Exit Function
Dim InputData As String
Dim Counter As Integer
Dim Counter2 As Integer
Dim InZone As Boolean
Counter = -1
Counter2 = -1
Open FileName For Input As #14
Do While Not EOF(14)
   Line Input #14, InputData
   If InZone Then
      If IsKey(InputData) Then
         Counter = Counter + 1
         If Left(InputData, Len(Key)) = Key Then
            GetKeyIndex2 = Counter
            Exit Do
         End If
      ElseIf IsSection(InputData) Then
         Exit Do
      End If
   Else
      If IsSection(InputData) Then Counter2 = Counter2 + 1
      If Counter2 = SectionIndexNum Then InZone = True
   End If
Loop
Close #14
End Function
```

EXHIBIT C

Exhibit C-1

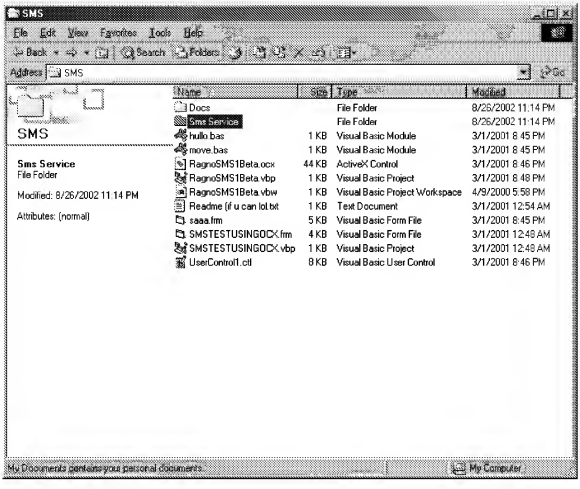

SMS
File Edit View Favorites Tools Help
Back Search Folders
Address SMS
Go
SMS
Sms Service
File Folder
Modified: 8/26/2002 11:14 PM
Attributes: (normal)
Name
Size
Type
Modified
Docs
File Folder
8/26/2002 11:14 PM
Sms Service
File Folder
8/26/2002 11:14 PM
hullo.bas
1 KB
Visual Basic Module
3/1/2001 8:45 PM
move.bas
1 KB
Visual Basic Module
3/1/2001 8:45 PM
RagnoSMS1Beta.ocx
44 KB
ActiveX Control
3/1/2001 8:46 PM
RagnoSMS1Beta.vbp
1 KB
Visual Basic Project
3/1/2001 8:48 PM
RagnoSMS1Beta.vbw
1 KB
Visual Basic Project Workspace
4/9/2000 5:58 PM
Readme (if u can lol.txt
1 KB
Text Document
3/1/2001 12:54 AM
saaa.frm
5 KB
Visual Basic Form File
3/1/2001 8:45 PM
SMSTESTUSINGOCX.frm
4 KB
Visual Basic Form File
3/1/2001 12:48 AM
SMSTESTUSINGOCX.vbp
1 KB
Visual Basic Project
3/1/2001 12:48 AM
UserControl1.ctl
8 KB
Visual Basic User Control
3/1/2001 8:46 PM
My Documents contains your personal documents.
My Computer

Exhibit C-2

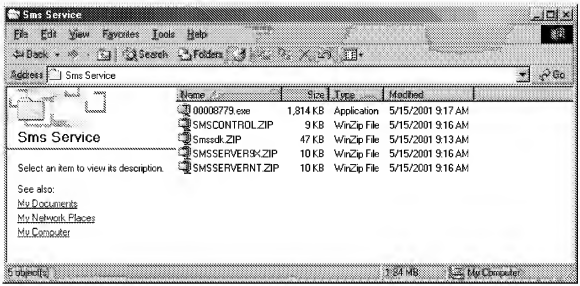

Sms Service
File Edit View Favorites Tools Help
Back Search Folders
Address Sms Service
Go
Sms Service
Select an item to view its description.
See also:
My Documents
My Network Places
My Computer
Name Size Type Modified
00008779.exe 1,814 KB Application 5/15/2001 9:17 AM
SMSCONTROL.ZIP 9 KB WinZip File 5/15/2001 9:16 AM
Smssdk.ZIP 47 KB WinZip File 5/15/2001 9:13 AM
SMSSERVER9X.ZIP 10 KB WinZip File 5/15/2001 9:16 AM
SMSSERVERNT.ZIP 10 KB WinZip File 5/15/2001 9:16 AM
5 object(s)
7.84 MB
My Computer

Exhibit C-3

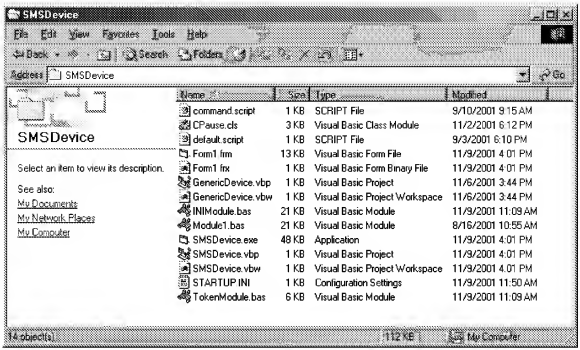

SMSDevice
File Edit View Favorites Tools Help
Back Search Folders
Address SMSDevice
Go
SMSDevice
Select an item to view its description.
See also:
My Documents
My Network Places
My Computer
Name Size Type Modified
command.script 1 KB SCRIPT File 9/10/2001 9:15 AM
CPause.cls 3 KB Visual Basic Class Module 11/2/2001 6:12 PM
default.script 1 KB SCRIPT File 9/3/2001 6:10 PM
Form1.frm 13 KB Visual Basic Form File 11/9/2001 4:01 PM
Form1.frx 1 KB Visual Basic Form Binary File 11/9/2001 4:01 PM
GenericDevice.vbp 1 KB Visual Basic Project 11/6/2001 3:44 PM
GenericDevice.vbw 1 KB Visual Basic Project Workspace 11/6/2001 3:44 PM
INIModule.bas 21 KB Visual Basic Module 11/9/2001 11:09 AM
Module1.bas 21 KB Visual Basic Module 8/16/2001 10:55 AM
SMSDevice.exe 48 KB Application 11/9/2001 4:01 PM
SMSDevice.vbp 1 KB Visual Basic Project 11/9/2001 4:01 PM
SMSDevice.vbw 1 KB Visual Basic Project Workspace 11/9/2001 4:01 PM
STARTUP.INI 1 KB Configuration Settings 11/9/2001 11:50 AM
TokenModule.bas 6 KB Visual Basic Module 11/9/2001 11:09 AM
14 object(s)
112 KB
My Computer

Exhibit C-4

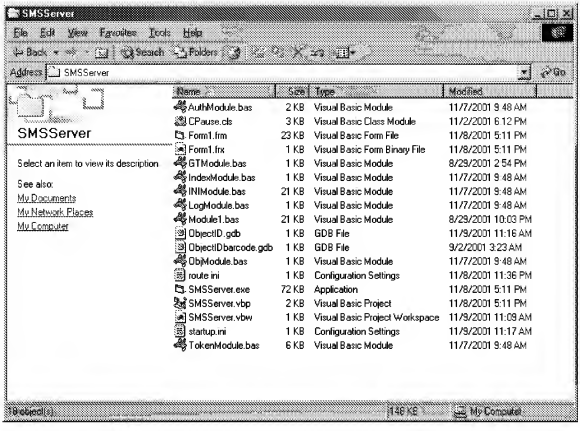

SMSServer
File Edit View Favorites Tools Help
Back Search Folders
Address SMSServer
Go
SMSServer
Select an item to view its description.
See also:
My Documents
My Network Places
My Computer
Name Size Type Modified
AuthModule.bas 2 KB Visual Basic Module 11/7/2001 9:48 AM
CPause.cls 3 KB Visual Basic Class Module 11/2/2001 6:12 PM
Form1.frm 23 KB Visual Basic Form File 11/8/2001 5:11 PM
Form1.frx 1 KB Visual Basic Form Binary File 11/8/2001 5:11 PM
GTModule.bas 1 KB Visual Basic Module 8/29/2001 2:54 PM
IndexModule.bas 1 KB Visual Basic Module 11/7/2001 9:48 AM
INIModule.bas 21 KB Visual Basic Module 11/7/2001 9:48 AM
LogModule.bas 1 KB Visual Basic Module 11/7/2001 9:48 AM
Module1.bas 21 KB Visual Basic Module 8/29/2001 10:03 PM
ObjectID.gdb 1 KB GDB File 11/9/2001 11:16 AM
ObjectIDbarcode.gdb 1 KB GDB File 9/2/2001 3:23 AM
ObjModule.bas 1 KB Visual Basic Module 11/7/2001 9:48 AM
route.ini 1 KB Configuration Settings 11/8/2001 11:36 PM
SMSServer.exe 72 KB Application 11/8/2001 5:11 PM
SMSServer.vbp 2 KB Visual Basic Project 11/8/2001 5:11 PM
SMSServer.vbw 1 KB Visual Basic Project Workspace 11/9/2001 11:09 AM
startup.ini 1 KB Configuration Settings 11/9/2001 11:17 AM
TokenModule.bas 6 KB Visual Basic Module 11/7/2001 9:48 AM
19 object(s)
148 KB
My Computer

Exhibit C-5

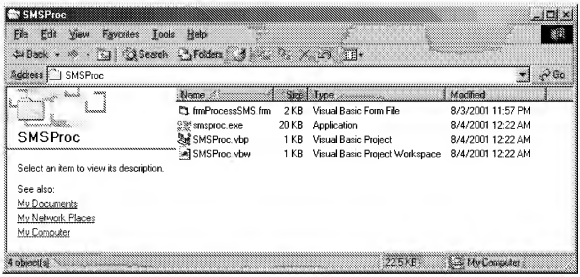

SMSProc
File Edit View Favorites Tools Help
Back Search Folders
Address SMSProc
Go
SMSProc
Select an item to view its description.
See also:
My Documents
My Network Places
My Computer
Name Size Type Modified
frmProcessSMS.frm 2 KB Visual Basic Form File 8/3/2001 11:57 PM
smsproc.exe 20 KB Application 8/4/2001 12:22 AM
SMSProc.vbp 1 KB Visual Basic Project 8/4/2001 12:22 AM
SMSProc.vbw 1 KB Visual Basic Project Workspace 8/4/2001 12:22 AM
4 object(s)
22.5 KB
My Computer

Exhibit C-6

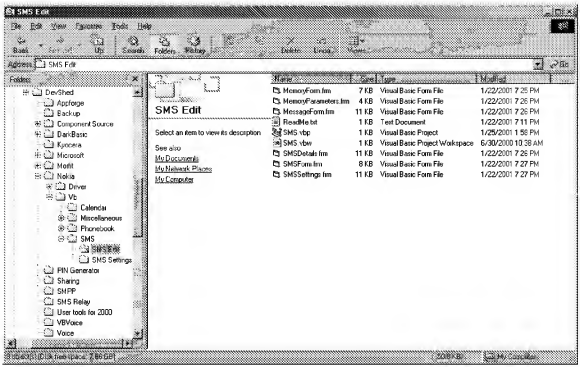
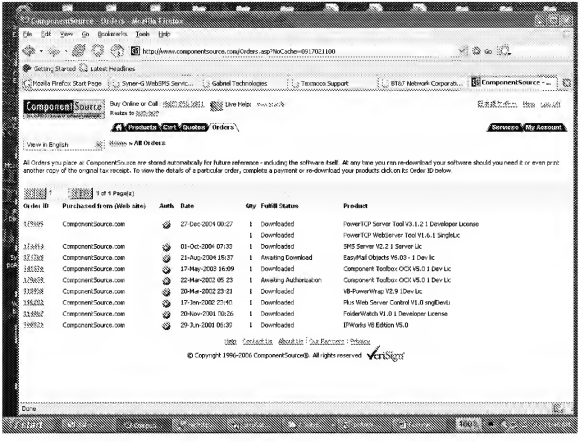

SMS Edit
File Edit View Favorites Tools Help
Back Up Search Folders History Delete Undo Views
Address SMS Edit
Folders
DevShed
Appforge
Backup
Component Source
DanBasic
Kyocera
Microsoft
Motif
Nokia
Driver
Vb
Calendar
Miscellaneous
Phonebook
SMS
SMS Edit
SMS Settings
PIN Generator
Sharing
SMPP
SMS Relay
User tools for 2000
VBVoice
Voice
SMS Edit
Select an item to view its description.
See also
My Documents
My Network Places
My Computer
Name Size Type Modified
MemoryForm.frm 7 KB Visual Basic Form File 1/22/2001 7:25 PM
MemoryParameters.frm 4 KB Visual Basic Form File 1/22/2001 7:25 PM
MessageForm.frm 11 KB Visual Basic Form File 1/22/2001 7:26 PM
ReadMe.txt 1 KB Text Document 1/22/2001 7:11 PM
SMS.vbp 1 KB Visual Basic Project 1/25/2001 1:56 PM
SMS.vbw 1 KB Visual Basic Project Workspace 6/30/2000 10:38 AM
SMSDetails.frm 11 KB Visual Basic Form File 1/22/2001 7:26 PM
SMSForm.frm 8 KB Visual Basic Form File 1/22/2001 7:27 PM
SMSSettings.frm 11 KB Visual Basic Form File 1/22/2001 7:27 PM

Exhibit C-7


ComponentSource - Orders - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.componentsource.com/Orders.asp?NoCache=6917021100
Getting Started Latest Headlines
Mozilla Firefox Start Page
Syner-G WebSMS Servic...
Gabriel Technologies
Tecnocos Support
BT67 Network Corporati...
ComponentSource - ...
ComponentSource
Buy Online or Call
Live Help
Products Cart Quotes Orders
Services My Account
View in English
Home > All Orders
All Orders you place at ComponentSource are stored automatically for future reference - including the software itself. At any time you can re-download your software should you need it or even print another copy of the original tax receipt. To view the details of a particular order, complete a payment or re-download your products click on its Order ID below.
1 of 1 Page(s)
Order ID
Purchased from (Web site)
Auth
Date
Qty
Fulfill Status
Product
178695 ComponentSource.com 27-Dec-2004 00:27 1 Downloaded PowerTCP Server Tool V3.1.2.1 Developer License
1 Downloaded PowerTCP WebServer Tool V1.6.1 SingleLic
174619 ComponentSource.com 01-Oct-2004 07:33 1 Downloaded SMS Server V2.2.1 Server Lic
171705 ComponentSource.com 21-Aug-2004 15:37 1 Awaiting Download EasyMail Objects V6.03 - 1 Dev lic
161576 ComponentSource.com 17-May-2003 16:09 1 Downloaded Component Toolbox OCX V5.0.1 Dev Lic
157658 ComponentSource.com 22-Mar-2002 05:23 1 Awaiting Authorisation Component Toolbox OCX V5.0.1 Dev Lic
138958 ComponentSource.com 20-Mar-2002 23:23 1 Downloaded VB-PowerWrap V2.9 1Dev Lic
136203 ComponentSource.com 17-Jan-2002 21:40 1 Downloaded Plus Web Server Control V1.0 singDevLi
134867 ComponentSource.com 20-Nov-2001 10:26 1 Downloaded FolderWatch V1.0 1 Developer License
130523 ComponentSource.com 29-Jun-2001 06:30 1 Downloaded IPWorks V8 Edition V5.0
Help Contact Us About Us Our Payment Privacy
© Copyright 1996-2006 ComponentSource®. All rights reserved
VeriSign
Done